LLNL-CONF-666690

On the Use of Finite Difference Matrix-Vector Products inNewton-Krylov Solvers for Implicit Climate Dynamics withSpectral Elements

C. S. Woodward, D. J. Gardner, K. J. Evans

January 30, 2015

International Conference on Computational Science
Reykjavík, Iceland
June 1, 2015 through June 3, 2015

Disclaimer

This document was prepared as an account of work sponsored by an agency of the United States government. Neither the United States government nor Lawrence Livermore National Security, LLC, nor any of their employees makes any warranty, expressed or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States government or Lawrence Livermore National Security, LLC. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States government or Lawrence Livermore National Security, LLC, and shall not be used for advertising or product endorsement purposes.

On the Use of Finite Difference Matrix-Vector Products in Newton-Krylov Solvers for Implicit Climate Dynamics with Spectral Elements

Carol S. Woodward[1], David J. Gardner[2], and Katherine J. Evans[3]

[1] Lawrence Livermore National Laboratory, Livermore, California U.S.A.
woodward6@llnl.gov
[2] Lawrence Livermore National Laboratory, Livermore, California U.S.A.
gardner48@llnl.gov
[3] Oak Ridge National Laboratory, Oak Ridge, Tennessee, U.S.A.
evanskj@ornl.gov

Abstract
Efficient solutions of global climate models require effectively handling disparate length and time scales. Implicit solution approaches allow time integration of the physical system with a step size governed by accuracy of the processes of interest rather than by stability of the fastest time scales present. Implicit approaches, however, require the solution of nonlinear systems within each time step. Usually, a Newton's method is applied to solve these systems. Each iteration of the Newton's method, in turn, requires the solution of a linear model of the nonlinear system. This model employs the Jacobian of the problem-defining nonlinear residual, but this Jacobian can be costly to form. If a Krylov linear solver is used for the solution of the linear system, the action of the Jacobian matrix on a given vector is required. In the case of spectral element methods, the Jacobian is not calculated but only implemented through matrix-vector products. The matrix-vector multiply can also be approximated by a finite difference approximation which may introduce inaccuracy in the overall nonlinear solver. In this paper, we review the advantages and disadvantages of finite difference approximations of these matrix-vector products for climate dynamics within the spectral element shallow water dynamical core of the Community Atmosphere Model (CAM).

Keywords: matrix-vector multiply, spectral element solvers, Newton's method

1 Introduction

Investigating global-scale climate dynamics demands simulations resolving atmospheric features with time scales covering multiple orders of magnitude. Dynamical cores, which encapsulate equations governing fluid dynamics in the atmosphere, commonly rely on explicit or semi-implicit methods for time integration (e.g. [9, 8]). While these schemes can be fast to implement, their time step size limitations present issues with accumulation errors when long simulations are required [16].

Implicit methods can deliver accuracy and stability for much larger time steps than semi-implicit or explicit methods. For an implicit approach to the fluid dynamics within an atmospheric dynamical core, a large-scale, nonlinear, algebraic system of equations must be solved within each time step. The fastest method for solving these nonlinear systems is a Newton method [7]. Each iteration of a Newton method in turn requires the solution of a linear, algebraic system of equations. Pairing a Newton method with a preconditioned Krylov iterative linear solver gives scalability for solving large-scale systems [17].

One interesting property of Krylov iterative methods is that at only one point in each iteration is information about the matrix required. In addition, that point only requires products of the matrix with a vector. In no part of the algorithm are actual matrix entries needed. In the case of a Newton-Krylov method, the linear system matrix is known to be the Jacobian of the nonlinear, system-defining function. Hence, these products can be approximated through a directional derivative of the nonlinear function. Prior work has shown that as long as the nonlinear function has sufficient continuity, and the finite difference is sufficiently accurate, then the overall Newton method will retain many of its positive properties [3, 2].

In the studies below we utilize the compatible and conservative spectral element method (SEM) dynamical core in the Community Atmosphere Model (CAM) [23]. This discretization technique is a continuous Galerkin hp finite element method where integrals in the weak formulation are evaluated using Gauss-Lobatto quadrature within each element. An orthogonal nodal basis is constructed from the tensor product of one-dimensional Lagrange interpolating polynomials defined on the Gauss-Legendre-Lobatto (GLL) nodes. SEM implementations leverage this tensor product formulation to efficiently compute the action of a matrix on a vector without explicitly forming the full matrix [5, 13, 12]. This matrix-free approach necessitates storing only one-dimensional element operators. As a result, for a d-dimensional problem with np GLL points along each dimension of an element, matrix-vector products can be computed for an element in $\mathcal{O}(np^{d+1})$ operations as opposed to $\mathcal{O}(np^{2d})$ operations for matrix-vector multiplication [5]. Both the analytic and finite difference matrix vector products take advantage of these efficient tensor product formulations. As such the Jacobian matrix is never assembled, and its information is re-computed for each Jacobian vector product.

The main advantage of using finite difference approximations to the matrix-vector product is the ease of implementation. When using this approximation, only evaluations of the nonlinear function are needed, and one already has that as part of the basic Newton method. No matrix elements need to be computed, so no matrix storage is required. In addition, expensive but non-dominant parts of the nonlinear function evaluation can either be lagged or approximated while still maintaining convergence of the overall Newton method [4]. Lastly, the finite difference approximation takes advantage of the most recent information about the nonlinear problem solution in each evaluation. Hence, it may offer an advantage in bringing new information into the solution process. The main disadvantages of using these finite difference approximations include a slowdown in the speed of the Newton method convergence, less robustness in the face of discontinuities in the function values (although Newton's method is no longer guaranteed to converge in this case), and increased costs over the analytic Jacobian when many iterations of the linear solver are needed for convergence.

In order to assess the relative impacts of these considerations on the convergence and efficiency of a climate simulation, in this paper, we present results of a numerical study comparing run times and iteration counts for various climate system test problems using the shallow water model in the spectral element dynamical core of CAM. The rest of the paper is organized as follows. In the next section, we overview the Newton-Krylov method and matrix-vector multiplication options. Section 3 shows results comparing the two matrix-vector multiply options, and Section 4 gives some conclusions from our study.

2 Methods

We perform our numerical studies in the context of the two-dimensional shallow water equations. These equations make a simple test bed that is representative of the full atmospheric model retaining many important physical scales. We let χ denote the collection of dependent variables written as a single quantity,

$$\chi = (v_1, v_2, p)^T . \tag{1}$$

Then, the shallow-water equations are

$$\frac{\partial \chi}{\partial t} = G(\chi), \tag{2}$$

where

$$G = \begin{pmatrix} -(\zeta + f)\hat{k} \times \mathbf{v} - \nabla(\frac{1}{2}\mathbf{v}\cdot\mathbf{v} + \Phi) \\ -\nabla \cdot (p\mathbf{v}) \end{pmatrix} \tag{3}$$

with initial conditions, where $\mathbf{v} = (v_1, v_2)^T$ is the velocity vector on the surface of the sphere, $p = gh$, where g is the gravitational acceleration constant, and h is the thickness of the fluid. The vorticity is $\zeta = \hat{k} \cdot \nabla \times \mathbf{v}$, with $\hat{k}$ representing the unit vector in the Cartesian z-plane. The Coriolis force is included as $f = 2\Omega \sin(\phi)$, based on the latitude ϕ. The geopotential height field is given by $\Phi = g(h + h_s)$ where h_s denotes the height of the underlying terrain, and Ω is the angular rotation of Earth.

As in [19], we employ a BDF2 second order implicit integration method which requires the solution of a nonlinear algebraic system at each time step. Given the approximate solution at time t^n, the approximate solution at time t^{n+1} is computed by finding the root of the nonlinear function, F, where

$$F(\chi) \equiv \frac{3}{2\Delta t}\left(\chi - \frac{4}{3}\chi^n + \frac{1}{3}\chi^{n-1}\right) - G(\chi) = 0. \tag{4}$$

We note that the choice of integrator does not affect results for our nonlinear solver.

To solve the nonlinear system at each time step, we employ a Newton-Krylov method. Newton's method is formulated from applying successive linear models of the nonlinear function (4). Each iteration requires finding $\delta\chi^k$ such that

$$J(\chi^k)\delta\chi^k = -F(\chi^k), \tag{5}$$

where the index, k, indicates the Newton iteration number, $J(\chi^k)$ is the Jacobian of F evaluated at χ^k, and $\delta\chi^k = \chi^{k+1} - \chi^k$.

A Krylov method is used to solve this linear system. Krylov methods only require information about the system matrix through the action of the matrix on a vector, $w = J(\chi)v$. As stated above, in the SEM these products can be efficiently computed per element in $\mathcal{O}(np^3)$ operations for the two-dimensional shallow water equations using tensor products of one-dimensional element operators. Specifically, the Jacobian of the residual equation requires the SEM discrete divergence, gradient, and curl operators which are constructed from the one-dimensional spectral differentiation matrix. The Jacobian-vector multiply is formulated by linearizing the undiscretized residual equation F and, for each element, using the discrete SEM operators to compute the necessary values at each of the GLL nodes. In this matrix-free approach Jacobian information is re-evaluated at each iteration of the Krylov method rather than assembling a matrix for reuse at the start of each Newton iteration.

Alternatively, we can exploit the fact that the system matrix is the Jacobian of the nonlinear function. In this case, the matrix-vector product can be approximated through a finite difference,

$$J(\chi)v \approx \frac{F(\chi + \epsilon v) - F(\chi)}{\epsilon}. \tag{6}$$

This inexact Newton-Krylov method avoids calculation of the matrix entries [3] but may slow convergence since it introduces an error into the effect of the Jacobian. That error is related to the size of ϵ [2]. Most implementations of Newton-Krylov methods use an ϵ based on the relative sizes of $\|\chi\|$ and $\|v\|$ as well as on the square root of unit roundoff [3, 17, 6]. Although, other authors have used variations on the choice of ϵ, the relative cost of the resulting solutions do not change significantly once the value is small [1]. We use the Newton-Krylov implementation in the Trilinos library which chooses ϵ such that $\epsilon = \lambda(\lambda + \frac{\|\chi\|}{\|v\|})$, where the default $\lambda = 1.0e-6$. If $\|\chi\|$ and $\|v\|$ are of similar sizes, then this value will be a little larger than the square root of unit roundoff for double precision. Below, we study the effect of changing the size of λ and hence ϵ.

In our work, the linear system is solved via Flexible GMRES [21] to allow for the preconditioner to change from one linear iteration to the next. We determine convergence for Newton's method by testing whether the relative residual of the k^{th} iterate has decreased sufficiently as quantified by a tolerance τ

$$\frac{\|F(\chi^k)\|_2}{\|F(\chi^0)\|_2} < \tau. \tag{7}$$

In (7), $F(\chi^k)$ is the nonlinear residual at step k, and the initial iterate, χ^0, is the solution at the previous time step. FGMRES is given an initial iterate $\chi = 0$, and we iterate until the relative linear residual at step m, is less than a specified tolerance η

$$\frac{\|J(\chi^k)\delta\chi^m + F(\chi^k)\|_2}{\|F(\chi^k)\|_2} < \eta. \tag{8}$$

Our implicit solution approach makes use of a block preconditioner outlined in [19]. Each call to this preconditioner employs subsidiary solves with GMRES. With a tight tolerance on these solves, the preconditioner results in a very effective solve using a minimal number of outer FGMRES iterations. With a looser tolerance, more FGMRES iterations are needed. In work below, we consider both loose and tight tolerances for the preconditioner GMRES iteration.

In considering the choice of using the finite difference or analytic matrix-vector product evaluations, we want to make some comments on cost. We first observe that the function is evaluated at the current solution iterate at the start of each Newton iteration. As a result, only one function evaluation is required for each linear iteration for the finite difference approximation. That function evaluation requires the computation of N entries where N is the number of unknowns in the problem. So, the cost of this approximation will increase as the number of linear iterations increases. At the continuous level, the Jacobian function has more terms than the nonlinear function. As a result the Jacobian-vector product using the analytic form of the Jacobian will require more floating point operations per evaluation and hence be more expensive. If the number of linear iterations is the same or if the finite difference approximation requires only a small increase in the number of iterations, we would expect the finite difference method to be faster.

Another consideration is that the finite difference matrix-vector multiply introduces an error term proportional to ϵ^2. When ϵ is small, this term likely does not affect results. However, for larger values of ϵ the error pollutes the linear model, and, thus, we might expect to see requirements for more linear iterations as the error will lead to a potentially less effective update to the nonlinear model.

Lastly, since the finite difference approach evaluates the nonlinear function at each iteration, it is incorporating information on the full nonlinear function evaluated with the newest information about the approximate solution. (Note that $\chi^k + \epsilon v$ gives a perturbation of the last iterate solution in the direction of the newest approximation.) As a result, for problems with significant nonlinearities, we may expect to see some benefit to the nonlinear solution process with this approximation.

3 Numerical Studies

In this section, we investigate the comparative performance of matrix-vector multiplications within the Newton-Krylov method.

3.1 Test Cases

As mentioned above, the cases included here are solved within the shallow water spectral element-based dynamical core option within the Community Atmosphere Model (CAM) [20]. The first case comes from a suite of canonical tests developed by Williamson et al. [24] and has already been discussed in the context of the implicit solution framework [11]. This case, 'TC5', mimics a realistic atmospheric flow scenario and exercises the full shallow water equations. Here, an initially smooth zonal flow and geopotential height field is perturbed by an isolated mountain feature in the surface topography. The system solves the full shallow water equation set and is fully nonlinear, requiring multiple Newton iterations per time step. We also consider a regionally refined version of this test; it has a coarser grid than the uniform case overall, but it is refined locally in a circular region around the mountain as explained in [15]. The second case, 'SJTC1', is a more recent test problem developed by Galewsky et al. [14] to explore the ability of the model to capture a clear separation in scales between the fast gravity waves and longer term vorticity dynamics. This case is initialized with a flow perturbation within a barotropically unstable midlatitude jet. Prior work has demonstrated that implicit solvers successfully step over the small-scale gravity waves yet capture the vortical structures accurately [16]. For all cases, we verified the L^2 norm of the approximate solution error compared to a reference solution (generated with an explicit method running a 0.1s time step) to be insensitive to the choice of matrix-vector multiply, including the differencing parameter, λ.

3.2 Implementation

To solve the shallow water equations on the sphere CAM uses a cubed-sphere grid based on a central projection [22]. The total number of elements in the discretization is $6ne^2$ and each element contains np^2 GLL nodes. For TC5 we use a 1/4 degree resolution and discretize with $np = 4$ and $ne = 120$ except in the regionally refined case which starts with 1.25 degree resolution and goes down to 1/8 degree. This case has 8 levels of refinement from $ne = 24$ to $ne = 240$. The refined case has 2,990 elements across the globe, as compared to 86,400 for the uniform case. With SJTC1, $np = 8$ and $ne = 24$ for all runs.

The implicit solver is built within CAM using an interface to the Trilinos software library [10, 18] providing Newton's method, Flexible GMRES as the linear Krylov solver, and an Epetra interface for matrix and vector operations performed by a block preconditioner [19]. These operations work with subroutines that apply the discrete spectral element operators in CAM and vector kernels and solvers in Trilinos. We use Flexible GMRES with a 50 vector basis to accommodate a preconditioner enveloping an iterative method itself. The nonlinear tolerance used, τ, was $1e-6$, and the FGMRES tolerance, η, was $1e-2$.

All runs were performed on the Livermore Computing Facility computer, Cab, where each node contains a total of 16 Intel Xeon 2.6 GHz cores. For tests simulating 1 day, runs utilized 16 MPI processes on 1 compute node. The 6 day SJTC1 case was run on 2 compute nodes with 32 MPI processes and the 15 day TC5 case used 4 compute nodes with 64 MPI processes. In all setups, the maximum wall clock time was set at 16 hours.

Run times are shown below for the nonlinear function evaluation and the matrix-vector multiply. In all cases, these timings are only for the function evaluations that appear in Newton's method and the finite difference approximation within the Jacobian-vector products in FGMRES. It should be noted that if the FGMRES linear solver reaches 50 iterations, it will return to the Newton method. The maximum allowed number of Newton iterations was 50 and if that number was attained, the run halted with a failure. Failure is also indicated if the run did not complete within 16 hours of wall clock time.

3.3 Numerical Results

Table 1 shows iteration counts for the TC5 case. We look at counts for a 1 day run with the preconditioner GMRES tolerance set to $1e-4$, denoted as 'PC1', and $1e-8$, denoted as 'PC2.' These are given for time step sizes of 1,200 s and 1,800 s. For the smaller time step, there is little difference in the results between PC1 and PC2. In both cases, we observe a lack of robustness with a large value of λ in the finite difference approach. Because the finite difference parameter is large, the error in the approximate matrix-vector multiply results in the linear solver getting a solution to a poor linear model of the nonlinear system. As a result, the Newton solver will converge slowly or not at all. In all cases we see that fewer nonlinear iterations are required for solution with the finite difference approach with small λ than with the analytic Jacobian. We believe these smaller numbers result from the new information brought into the linear solve with the new function evaluations. Hence, a better linear model is obtained for the newest nonlinear iterate, and the Newton solver can make a bit more progress with the linear solution.

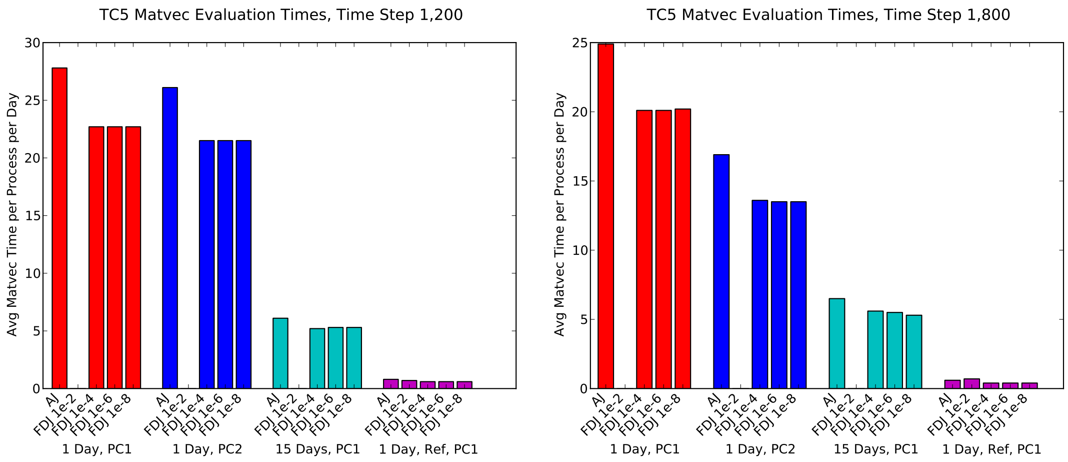
We also show iteration counts for the 15 day run and for the regionally refined run. The regionally refined run requires fewer linear iterations to solve the larger time step problem due to its smaller size. In addition, due to the application of refinement around the areas of greatest change, the nonlinear problem may appear numerically smoother and result in an easier nonlinear problem. Similar to the 1 day tests, the 15 day runs demonstrate concerns with robustness of the finite difference approach with too large of a difference parameter. Additionally, with the larger time step in the 15 day run and with a large λ value in the regionally refined case, more nonlinear iterations are required for the finite difference than with the analytic Jacobian. This larger number results from error in the finite difference scheme polluting the linear solution enough to result in a poor Newton update. In general, we see little difference between $\lambda = 1e-6$ and $\lambda = 1e-8$. Given the potential for loss of robustness, the $1e-8$ value is likely to be preferred, but tests on the next case help to verify this.

Figure 1 shows the total time taken in both the nonlinear function evaluation and matrix-vector products for the both the analytic and finite difference approaches for TC5 with both

Δt (s)	Jv	λ	1 day, PC1		1 day, PC2		15 days, PC1		1 day, ref, PC1	
			NNI	ALI	NNI	ALI	NNI	ALI	NNI	ALI
1,200	AJ		276	2.2	264	2.1	3,319	2.4	229	2.3
	FD	1e-2	Failed		Failed		Failed		234	2.5
		1e-4	264	2.2	255	2.1	3,307	2.4	219	2.3
		1e-6	264	2.2	254	2.1	3,307	2.4	219	2.3
		1e-8	264	2.2	254	2.1	3,308	2.4	219	2.3
1,800	AJ		201	3.2	160	2.3	2,723	3.8	155	2.6
	FD	1e-2	Failed		Failed		Failed		159	5.1
		1e-4	190	3.2	150	2.4	2,733	3.9	144	2.7
		1e-6	190	3.2	149	2.4	2,711	3.8	144	2.7
		1e-8	190	3.2	149	2.4	2,710	3.8	144	2.7

Table 1: Total nonlinear (NNI) and average linear (ALI) iteration counts for TC5 run for 1 day with preconditioner tolerances of $1e-4$ (PC1) and $1e-8$ (PC2), for 15 days with PC1, and for 1 day with PC1 and regional refinement. We see greater robustness with the analytic Jacobian (AJ) and that $\lambda < 1e-4$ gives the best performance for the finite difference (FD) approximation.

time step sizes. The results are the average time for each MPI process and in the 15 day run the results are also averaged per simulated day. In nearly all cases, the times show a definite speed benefit from using the finite difference approach. Although, the difference in timing between the analytic and finite difference approaches is less pronounced in the 15 day and regionally refined cases.



(a) $\Delta t = 1,200$

(b) $\Delta t = 1,800$

Figure 1: Total time taken (averaged over number of processing cores) in the matrix-vector product and function evaluation for TC5 run for 1 day with preconditioner tolerances of $1e-4$ (PC1) and $1e-8$ (PC2), for 15 days (average per simulated day) with PC1, and for 1 day with PC1 and regional refinement. In all cases the finite difference approximation out performs the analytic approach.

Table 2 shows iteration counts for the SJTC1 test case again for $1,200$ s and $1,800$ s time steps and with the preconditioner GMRES tolerance set to two values, $1e-4$, denoted as 'PC1', and $1e-8$, denoted as 'PC2.' We again see a loss of robustness with the finite difference approximation. This problem is more nonlinear than the TC5 case, so we expect that errors in the linear solution will certainly impede nonlinear solution progress more. We see this requirement for accuracy in that the numbers of nonlinear iterations are higher for $\lambda > 1e-8$ whereas $\lambda = 1e-6$ worked well for TC5.

Δt (s)	Method	λ	1 day, PC1		1 day, PC2		6 days, PC1	
			NNI	ALI	NNI	ALI	NNI	ALI
1,200	AJ		238	3.2	234	2.5	1,439	6.1
	FD	1e-2	Failed		Failed		Failed	
		1e-4	322	8.6	234	2.7	Failed	
		1e-6	229	3.3	221	2.6	1,465	6.3
		1e-8	228	3.3	221	2.6	1,428	6.2
1,800	AJ		197	27.9	161	5.8	1,168	35.0
	FD	1e-2	Failed		Failed		Failed	
		1e-4	496	42.7	Failed		4,565	46.7
		1e-6	207	30.5	148	6.3	1,354	37.0
		1e-8	185	27.7	148	6.4	1,158	35.1

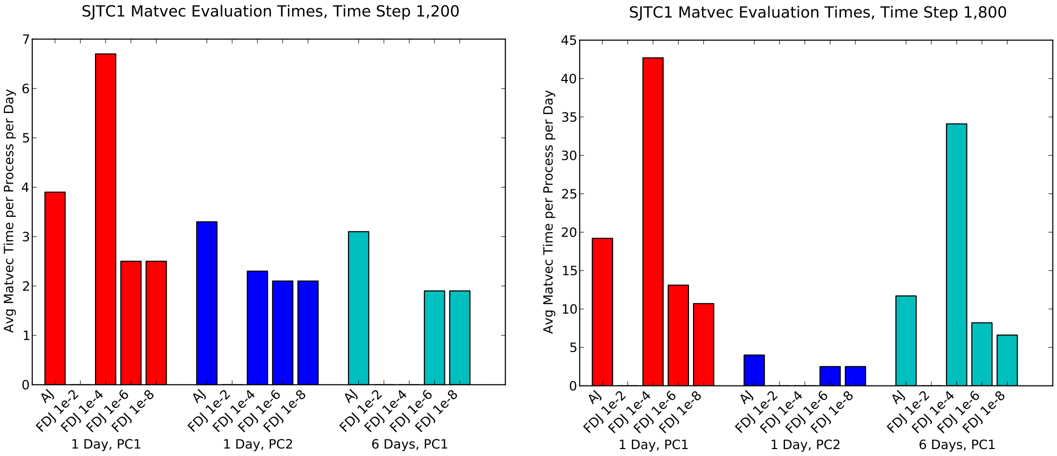
Table 2: Total nonlinear (NNI) and average linear (ALI) iteration counts for SJTC1 run for 1 day with a preconditioner tolerance of $1e-4$ (PC1) and $1e-8$ (PC2) and for 6 days with PC1. Note the requirement of $\lambda < 1e-6$ for dependable benefit of the finite difference approach.

Figure 2 shows the total time taken per MPI process in both the nonlinear function evaluation and matrix-vector products for the both the analytic and finite difference approaches for SJTC1. The 6 day run times are per simulated day. Unlike with TC5, we see increased times with the finite difference method for large λ compared to the analytic approach. As λ is decreased the run times improve and again a speed benefit is realized with the finite difference method. For the larger time step we also see that $\lambda = 1e-8$ gives the shortest run times.

4 Conclusions

From these results we can make a number of conclusions. First, too large a value of λ in the finite difference approximation leads to a loss of robustness. Both solver failures and significant slowdowns were observed due to the error pollution from the difference approximation. Second, a λ value of $1e-8$ should be used. The standard default of $1e-6$ was not small enough to give the maximum benefit for the more nonlinear problem with the larger time step. As we move toward larger step sizes, this benefit will translate to faster run times.

For spectral element methods, typically little Jacobian information is precomputed at the start of each Newton iteration. In this case, the finite difference approach will generally be faster, as is seen in this paper. Future work will explore the viability of precomputing Jacobian information and using this within the analytic Jacobian-vector product. This optimization may significantly affect the relative benefit of the two approaches. The tradeoff in this approach is the added complexity of implementation and increased storage requirements. However, a further significant benefit is the exploitation of this precomputed information within the preconditioner to further improve performance.



(a) $\Delta t = 1,200$ (b) $\Delta t = 1,800$

Figure 2: Total time taken (averaged over number of processing cores) in the matrix-vector product and function evaluation for SJTC1 run for 1 day with preconditioner tolerances of $1e-4$ (PC1) and $1e-8$ (PC2), for 6 days (average per simulated day) with PC1, and for 1 day with PC1 and regional refinement. Note again the smaller times for the finite difference approximation once the difference parameter is small.

Acknowledgements

The authors wish to thank Mark Taylor of Sandia National Laboratories (SNL) for instruction on the use of the regionally refined TC5 test case and Andy Salinger of SNL for his assistance with utilizing the PIRO interface in Trilinos. Support for this work was provided through the Scientific Discovery through Advanced Computing (SciDAC) program funded by the U.S. Department of Energy Office of Advanced Scientific Computing Research and the Office of Biological and Environmental Research. This work was partially performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344. The submitted manuscript is based upon work, authored in part by contractors [UT-Battelle LLC, manager of Oak Ridge National Laboratory (ORNL)]. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.

References

[1] H.-B. An, J. Wen, and T. Feng. On finite difference approximation of a matrix-vector product in the Jacobian-free Newton–Krylov method. *Journal of Computational and Applied Mathematics*, 236(6):1399–1409, 2011.

[2] P. N. Brown. A local convergence theory for combined inexact-Newton/finite-difference projection methods. *SIAM J. Numer. Anal.*, 24:407–434, 1987.

[3] P. N. Brown and Y. Saad. Hybrid Krylov methods for nonlinear systems. *SIAM Journal on Scientific and Statistical Computing*, 11(3):450–481, 1990.

[4] P. N. Brown, H. F. Walker, R. Wasyk, and C. S. Woodward. On using approximate finite differences in matrix-free Newton-Krylov methods. *SIAM J. Numer. Anal.*, 46(4):1892–1911, 2008.

[5] C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. Zang. *Spectral Methods: Fundamentals in Single Domains*. Springer, 2006.

[6] A. M. Collier, A. C. Hindmarsh, R. Serban, and C. S. Woodward. User documentation for KINSOL v2.7.0. Technical Report UCRL-SM-208116, Lawrence Livermore National Laboratory, 2012.

[7] J. E. Dennis and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. SIAM, Philadelphia, PA, 1996.

[8] ECMWF. IFS Documentation - Cy38r1, Part III: Dynamics and Numerical procedures. Technical Report cy38ra, European Centre for Medium-Range Weather Forecasting, June 2012.

[9] K. Evans, P. Lauritzen, R. Neale, S. Mishra, M. A. Taylor, and J. Tribbia. AMIP simulation with the CAM4 spectral element dynamical core. *Journal of Climate*, 26:689–709, 2013.

[10] K. J. Evans, D. W. I. Rouson, A. G. Salinger, M. A. Taylor, W. Weijer, and J. W. III. *A Scalable and Adaptable Solution Framework within Components of the Community Climate System Model*, volume 5545 of *Lecture Notes in Computer Science*, pages 332–341. Springer, 2009.

[11] K. J. Evans, M. Taylor, and J. B. Drake. Accuracy analysis of a spectral element atmospheric model using a fully implicit solution framework. *Monthly Weather Review*, 138:3333–3341, 2010.

[12] P. Fischer and J. Lottes. Hybrid schwarz-multigrid methods for the spectral element method: Extensions to navier-stokes. In T. Barth and et al., editors, *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 35–49. Springer Berlin Heidelberg, 2005.

[13] P. F. Fischer. An overlapping schwarz method for spectral element solution of the incompressible navier-stokes equations. *Journal of Computational Physics*, 133:84–101, 1997.

[14] J. Galewsky, R. K. Scott, and L. M. Polvani. An initial-value problem for testing numerical models of the global shallow-water equations. *Tellus*, 56(A):429–440, 2004.

[15] O. Guba, M. Taylor, P. Ullrich, J. Overfelt, and M. Levy. The spectral element method on variable resolution grids: evaluating grid sensitivity and resolution aware numerical viscosity. *Geophys. Model Dev.*, 7:4081–4117, 2014.

[16] J. Jia, J. C. Hill, K. J. Evans, G. I. Fann, and M. A. Taylor. A spectral deferred correction method applied to the shallow water equations on a sphere. *Monthly Weather Review*, 141:3435–3449, 2013.

[17] D. A. Knoll and D. E. Keyes. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *J. Comp. Phys.*, 193:357–397, 2004.

[18] P. A. Lott, H. C. Elman, K. J. Evans, X. S. Li, A. G. Salinger, and C. S. Woodward. Recent progress in nonlinear and linear solvers. In *SciDAC*, Denver, CO July 10-14, 2011, URL http://www.mcs.anl.gov/uploads/cels/papers/scidac11/final/lottaaron.pdf., July 2011.

[19] P. A. Lott, C. S. Woodward, and K. J. Evans. Algorithmically scalable block preconditioner for fully implicit shallow-water equations in CAM-SE. *Computational Geosciences*, 2014. DOI: 10.1007/s10596-014-9447-6.

[20] R. B. Neale, et al. Description of the Community Atmosphere Model (CAM 5.0). *NCAR Technical Note*, 2012. TN-486+STR.

[21] Y. Saad. A flexible inner-outer preconditioned GMRES algorithm. *SIAM Journal on Scientific Computing*, 14(2):461–469, 1993.

[22] R. Sadourny. Conservative finite-difference approximations of the primitive equations on quasi-uniform spherical grids. *Monthly Weather Review*, 100(2):136–144, 1972.

[23] M. A. Taylor and A. Fournier. A compatible and conservative spectral element method on unstructured grids. *Journal of Computational Physics*, 229:5879–5895, 2010.

[24] D. L. Williamson, J. B. Drake, J. J. Hack, R. Jakob, and P. N. Swarztrauber. A standard test set for numerical approximations to the shallow water equations in spherical geometry. *Journal of Computational Physics*, 102(1):211–224, 1992.